CAAP Quarterly Report

Date of Report: *April 7, 2020*

Prepared for: *Thomas Finch (Project Manager) and Joshua Arnold (CAAP Program Manager), U.S. DOT Pipeline and Hazardous Materials Safety Administration*

Contract Number: *693JK31850005CAAP*

Project Title: *Low-variance Deep Graph Learning for Predictive Pipeline Assessment with Interacting Threats*

Prepared by: *Hao Zhang (Colorado School of Mines) and Yiming Deng (Michigan State University)*

Contact Information: *Dr. Hao Zhang, Department of Computer Science, Colorado School of Mines; 1500 Illinois St., Golden, CO 80401; Phone: 303-273-3581; Email: hzhang@mines.edu*

For quarterly period ending: *April 7, 2020*

Business and Activity Section

(a) Contract Activity

No contract modification was made or proposed in this quarterly period. No materials were purchased during this quarterly period.

(b) Status Update of Past Quarter Activities

(c) Cost Share Activity

PI Zhang used his 11.29% yearly effort as the in-kind cost share to work on the project at the Colorado School of Mines. Co-PI Yiming Deng used his 6.07% yearly effort as the in-kind cost share to work on the project at the Michigan State University. The cost share was used following the approved proposal and no modification was made.

(d) Performed Research: Developing and Evaluating New Methods for Low-Variance Interacting Threats Assessment

1. Progress on Task 3: Deep Learning Based Threats Characterization and Analysis (MSU)

1.1 Simulation Environment Specification

The goal of current simulation is to establish the link between multi-NDE sensing measurements and defect profile (i.e., the size and depth detection and characterization), which will help us to develop probabilistic models of failure pressure of a pipeline containing defects, achieving predictions that are unbiased with reduced variability. In the past few quarters, we have done our FEM study for defect detection by Magnetic flux leakage (MFL), Pulse Eddy Current (PEC) and their fused models. In this quarter we have developed our defect detection scheme based on Eddy current (EC) based NDE technique. The fundamental problem is to determine according to signal shape whether is some defect, structural element, roughness or impurity inside the pipeline. Like before we have considered the pipeline specimen to be a flat 2D surface with flat bottom hole defects embedded within the specimen. Numerical data (i.e., FE analysis results) in our study should also be well managed as well as specifying simulation environment to be better compared with future experimental results. EC probe and the constructed model specifications are listed as follows:

Table 1: Eddy Current simulation specification

Specification	**Value**
Magnetic field Frequency domain Study with Coil Geometry Analysis has been involved here	
For the Coil Geometry Analysis study, the involved equations are:	$\nabla \times \vec{H} = \vec{J}, B = \nabla \times A,$ $J = \sigma E$
Coil Type	Numeric
Coil excitation	Current
Probe Types	Pancake coil, D coil probe
Coil Current	1 [A]
Number of turns in coil	50
Coil wire Conductivity	$6e7[S/m]$
Coil wire cross-section area	coil_thickness^2/50[m^2]
Conductor model	Homogenized Multi turn
Coil thickness	0.25 [mm]
Coil liftoff	0.15[mm]
Coil Diameter	2 [mm]
Sample plate length	30 [mm]
Sample plate breadth	16 [mm]
Sample plate depth	6 [mm]
Air block length (for insulation)	$1.5 *$ Sample plate length
Air block breadth (for insulation)	$1.5 *$ Sample plate breadth
Air block height (for insulation)	$6 *$ Sample depth
Defect diameter	1.5[mm], 3[mm], 6[mm]
Defect depth	0.1-5[mm] (varied)
Operating Frequencies	$5\ kHz, 80kHz, 200kHz, 2MHz$

Mesh (Different for the sample, coils and defect)	Finer mesh

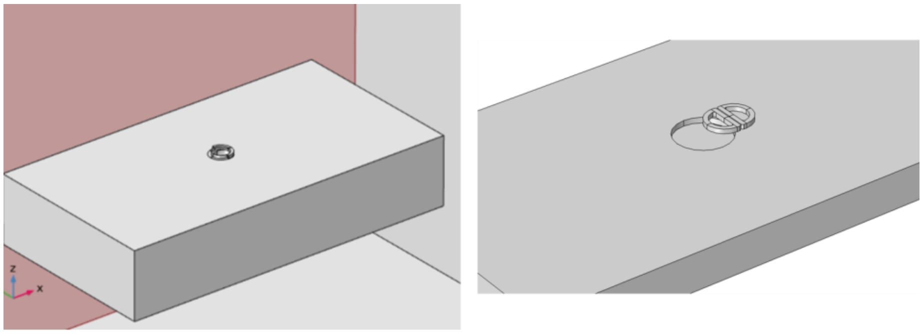
Here we have used two different coils for sensitivity analysis as transducers: pancake coil probe, D-coil probe. From the results of the line scans and 2D scans it is evident that D-coil differential probe has more sensitivity than the pancake coil. Here for defect characterization we have considered flat bottom hole defects of mainly three diameters which are being used in industry $1.5\ mm, 3\ mm\ and\ 6mm$. For each defect diameter we have studied the coil impedance in the range of line scans or 2D scans for various depths ranging from $0.1mm\ to\ 5mm$. All these studies are done in different frequency regimes of low to high frequency ranges of $5kHz, 80\ kHz, 200\ kHz, 2MHz$.

Low frequencies are more suitable for evaluation of identifiable anomalies at the outside wall of the tube, e g of the presence of construction elements of the steam generator. At high frequencies there are dominant signal components corresponding to the changes of internal tube wall, e g to the changes of profile tube [1]. Middle frequencies enable to obtain phase separation of 2D curves corresponding to defects that are important for quantification of percentual material drop. Here as we have taken the coil current to be 1 [A], hence the coil impedance and the coil voltage will eventually be the same here.

Table 2: Data form of EC results

Densities and distribution patterns of induced eddy currents	In matrix form
Readings from pic	Vector
Amplitude of the rising peak of differential induced voltage or the coil impedance	Scalar
For Pancake coil, $\Delta\ Coil\ impedance$	In matrix form
Differential form (for artificial defect) in D coils	
$\Delta\ Coil\ impedance$	In matrix form
$abs((\Delta\ Coil\ impedance)^2)$	In matrix form
$\Delta\ Coil\ Voltage$	In matrix form

1.2 Simulation Model Picture and Description



Pan cake coil above defect

D shaped coil above defect

Figure 1: Pan cake and D shaped coil acting as transducers

Here in the above picture the pancake and the D-coil probe is placed above the flat bottom hole defect. The below picture shows the model after application of the mesh.

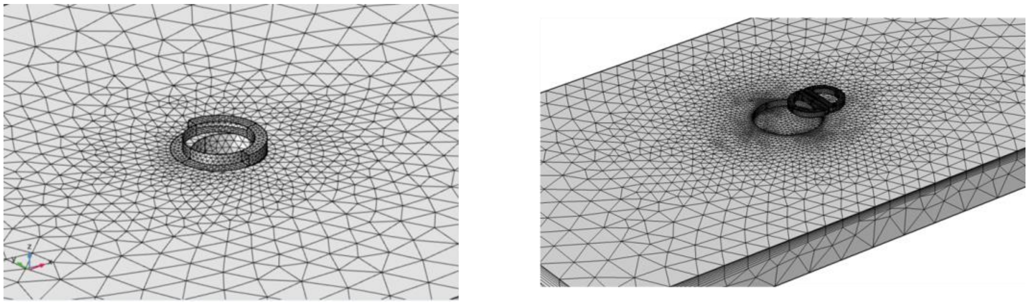


Figure 2: After applying the mesh

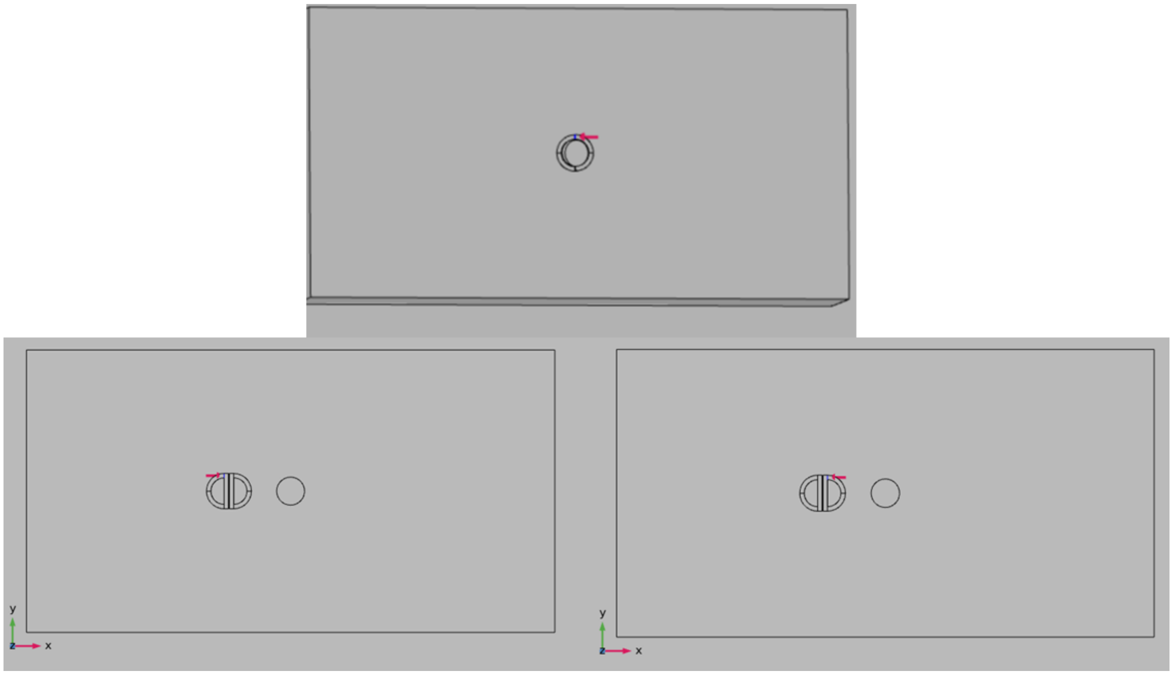
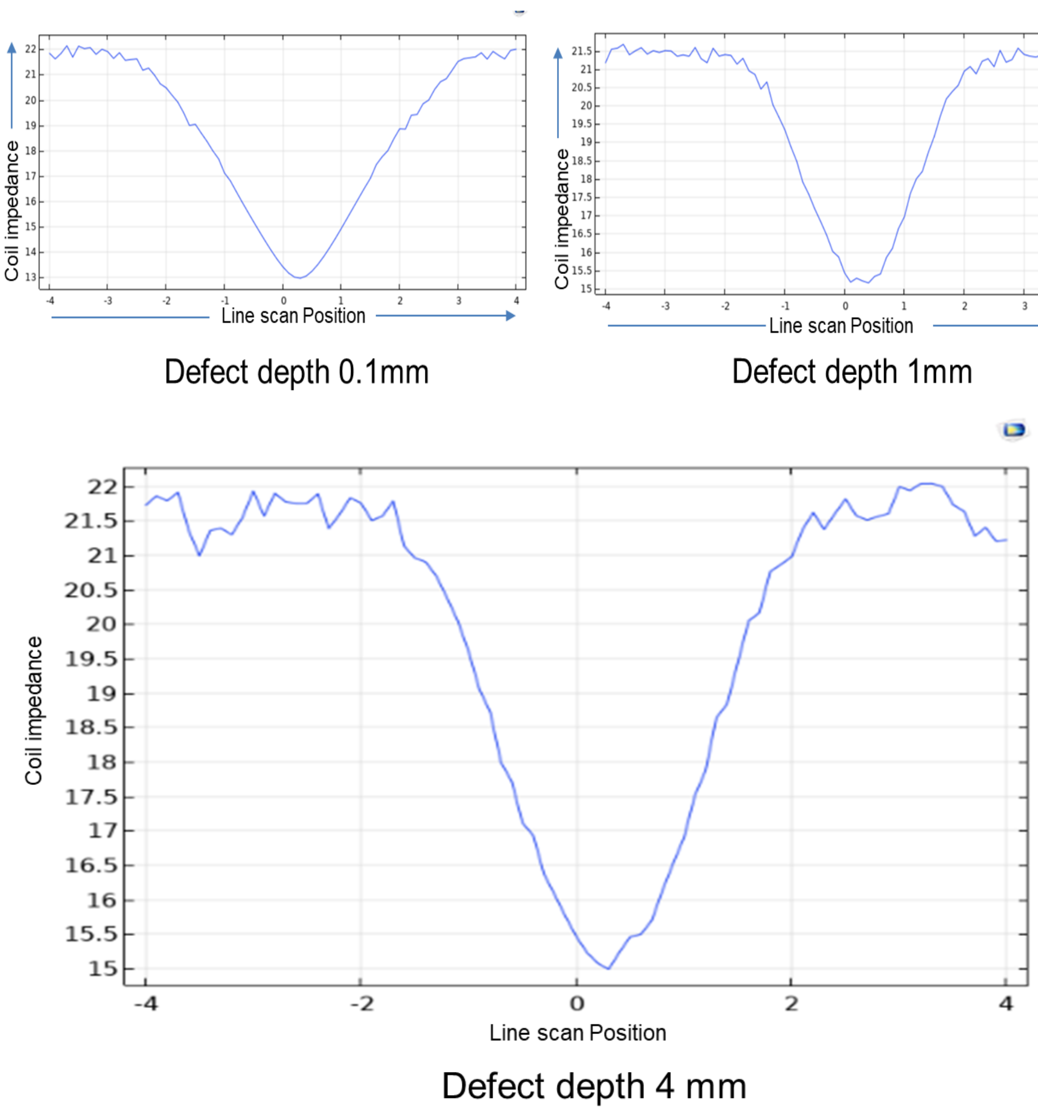


Figure 3: Current direction applied on the coils

From figure 3 it is evident that in D coils the direction of the current is in opposite direction in the two halves.

1.3 Results from the line scans of the Pan-Cake coil Probe

At first we have done the line scans using the pancake coil when the flat bottom hole defect has diameter of 1.59 mm and depth of of 0.1 mm, $1mm$ and 4.5 mm respectively at high frequency of 2 Mhz. Here we are using very fine step size of 0.1mm. The results of the coil impedance plot for the various depths are as follows:



Defect depth 0.1mm

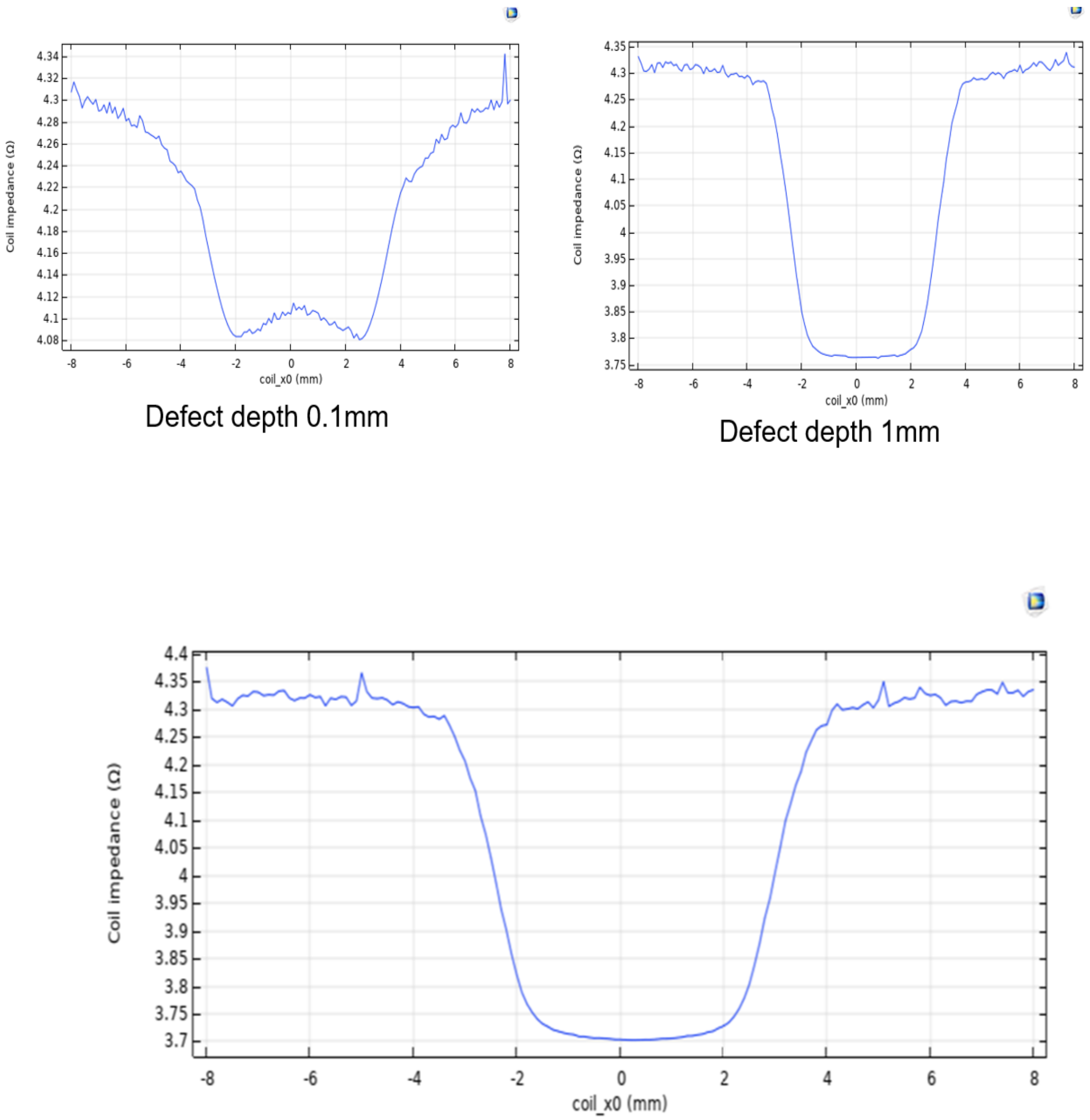
Defect depth 1mm

Defect depth 4 mm

Figure 4: Signals for a defect of dia 1.59 mm and of depths .1,1 and 4 mm respectively at operating freq of 2MHz

From the line scan results it is clear that with the change in the depths of the defect, there is change in the coil impedance.

Now , by changing the diameter of the defect to be 6 mm we are checking whether the transducer is sensitive enough to capture the changes in the coil impedance for various depths of the defects at an operating frequency of $200KHz$.



Defect depth 0.1mm

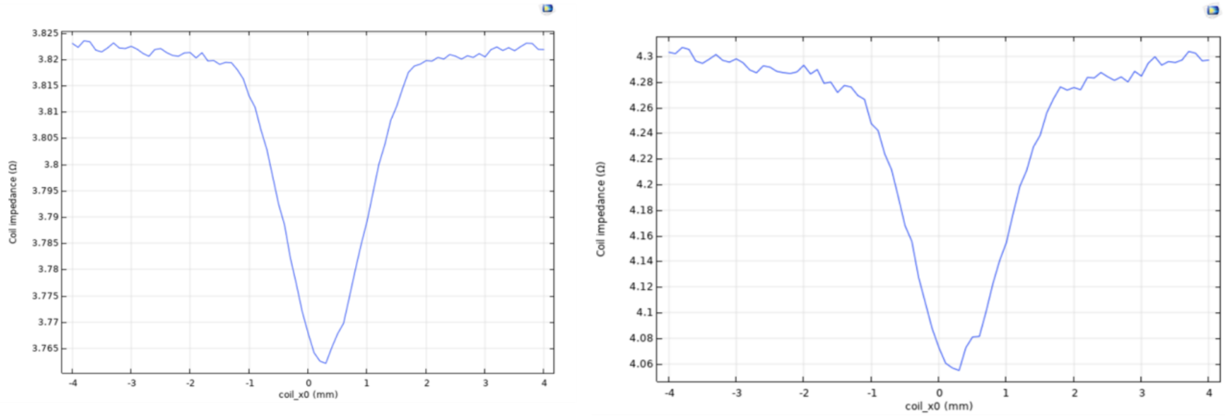
Defect depth 1mm

Defect depth 3 mm

Figure 5: Signals for a defect of dia 6 mm and of depths 0.1,1 and 3 mm respectively at operating freq of 2MHz

From the Figures 4 and 5, it is clearly evident that with the increase in diameter of the defects the signal changes i.e with more the diameter of the defect more is the spread in x direction. Moreover, we can conclude from Figure 5 that with the increase in depths of the defect the signal amplitude also increases.

Now on keeping the defect diameter to be constant at $1.59\ mm$ and the defect depth to be constant at $0.0125[in]$ and varying the frequency to $80kHz$ and $200\ kHz$. We observed from the below plot that intensity of the signal increases with increase in operating frequency.
In the below figure we can see that at $80kHz$ the signal ranges from 3.825- 3.765 whereas at a higher frequency of $200kHz$ it ranges from $4.3 - 4.06$.

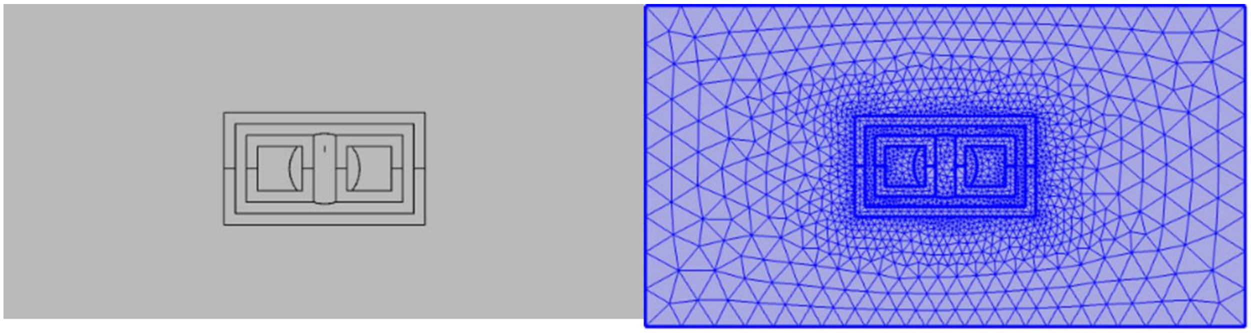


Operating freq: 80kHz

Operating freq: 200kHz

Figure 6: Coil impedance with the change in frequency

Now the differential coil probe is used as transducer in our model to increase the efficiency. The below figures show the basic model of the differential probe in COMSOL.

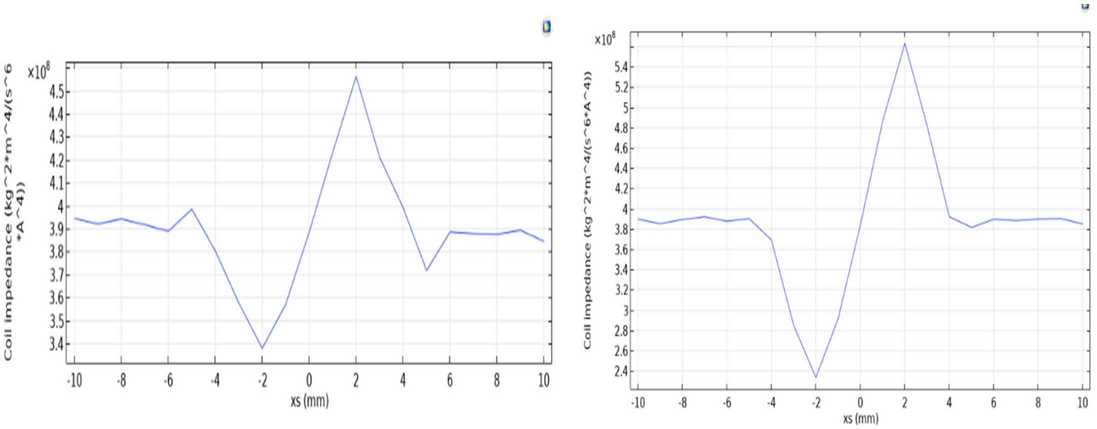


Differential Probe without mesh

Differential Probe with mesh

Figure 7: Differential probe design for EC analysis

Here we are evaluating the coil impedance as the absolute value of the square of the difference in the coil impedances against the line scan distance. For the first case for a defect of diameter $3.175\ mm$ and at an operating frequency of $200\ kHz$ we are evaluating the coil impedance for defect depths of $0.1\ \&\ 1\ mm$ respectively. From Figure 7 it is clearly evident that differential probe signal has much more sensitivity than pan cake coil signal. For the defect of depth $0.1mm$ the absolute value in the difference in coil impedance varies from $4.5 - 3.4$ peak to peak whereas for the $1mm$ defect it varies from $5.4 - 2.4$ peak to peak keeping other conditions fixed. From this we can conclude that differential probes can capture the defect depth information more accurately than the pancake coils.

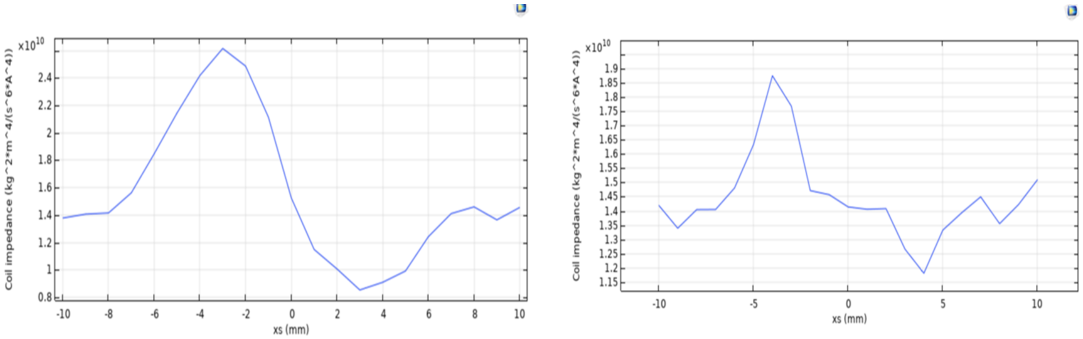


Defect depth 0.1mm

Defect depth 1mm

Figure 8: Varying depth analysis keeping defect diameter and operating frequency fixed

Now we have altered the frequency to $2MHz$ keeping other parameters same. With the change in frequency we observed the change in signal amplitude as follows.



Defect depth 0.1mm

Defect depth 1mm

Figure 9: Effect on the change in frequency

1.4 New Interacting defect design

Table 3. Simulation Setup

Sample plate length	30 [mm]
Sample plate breadth	16 [mm]
Sample plate depth	6 [mm]
Air block length (for insulation)	1.5 ∗ Sample plate length
Air block breadth (for insulation)	1.5 ∗ Sample plate breadth
Air block height (for insulation)	6 ∗ Sample depth
Surface defect diameter	1.5[mm], 3[mm], 6[mm]

Sub surface defect diameter	1.5[mm], 3[mm], 6[mm]
Sub surface defect depth	6 [mm]
Surface defect depth	0.1-5[mm] (varied)
Operating Frequencies	5 $kHz, 80kHz, 200kHz, 2MHz$
Mesh (Different for the sample, coils and defect)	Finer mesh

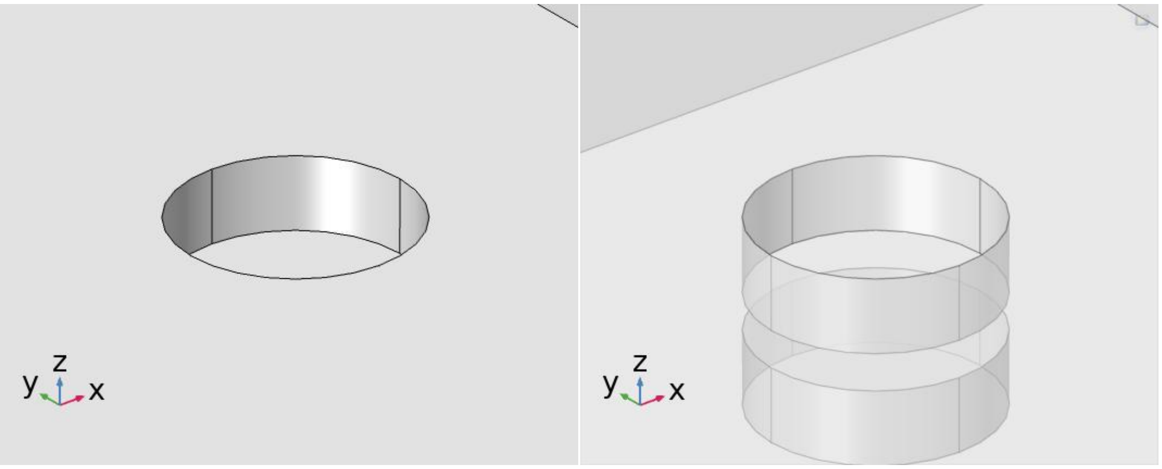
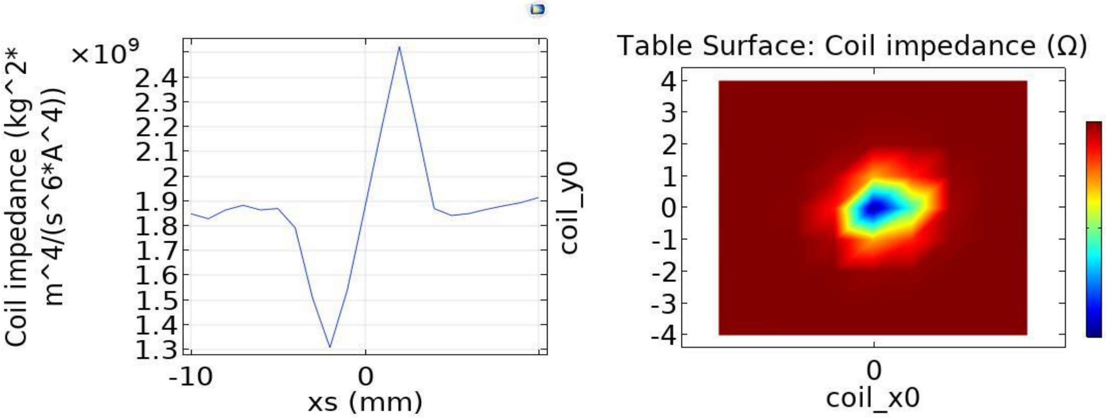
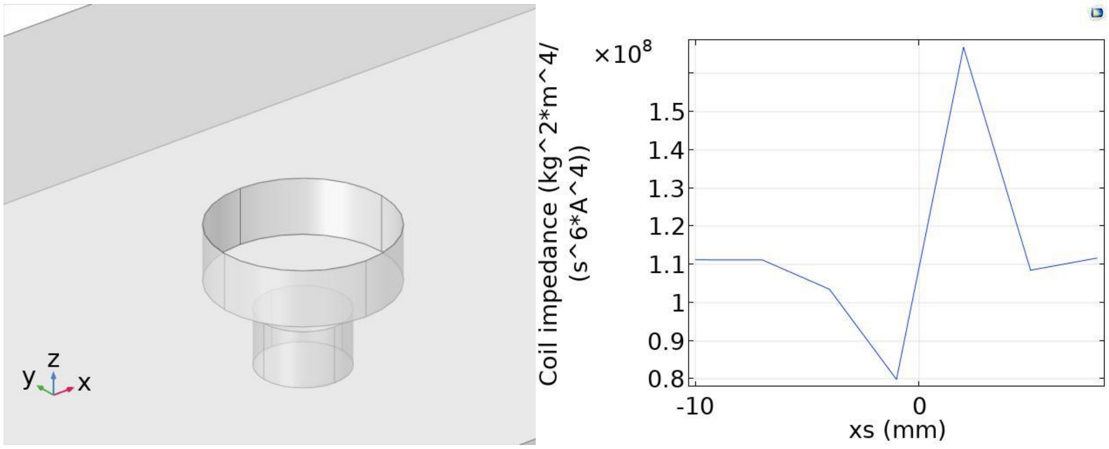


Figure 10: Surface and Sub surface defect location





Reference: Grman, J., R. Ravas, and L. Syrová. "Application of neural networks in multifrequency eddy-current testing." Measur. Sci. Rev 1, no. 1 (2001): 25-28.

2. Progress on Task 4: Predictive Interacting Threat Assessment (Mines)

In order to assess the pipeline integrity based on the detected interacting threats, we need to design a novel algorithm, which should consider the attribute of each threat (e.g., severity, size, etc.) and the distribution of the threats. In other word, we need to model the situation constructed by interacting threads, which is scale-variant (the number of threats is not fixed) and unstructured (the description of individual thread and their distribution is not unified). Thus, we propose to use graph to model interacting threats, in which the node of graphs denotes the attribute of the threat and the edge denotes the spatial distance between a pair of threats.

Given the graph-based modeling of interacting threats and by taking advantage of graph neural network (**GNN**), we can assess the failure probability of the pipeline based on the attribute and spatial distribution of interacting threats. In this report, we will mainly explain the theoretical and practical modeling of interacting threats with graphs, which is consistent with the timeline. For the predictive model (GNN), we will start to work on it in next quarter.

2.1 Interacting threats modeling by graphs

In this section, we will report on the preliminary of the graph-based input of GNN and then explain how we formulate and implement our pipeline assessment problem into a GNN model.

2.1.1. Preliminary

The recent success of neural network has boosted study on pattern recognition and data mining. Due to the effective capability of deep learning in capturing hidden pattern of Euclidean data, there is an increasing number of applications with graph-based data as input. GNN is designed for dealing with graph-based data in deep learning way. The purpose of GNN is to integrate the attributes and spatial distribution of nodes in graphs for prediction tasks, as shown in Figure 11.

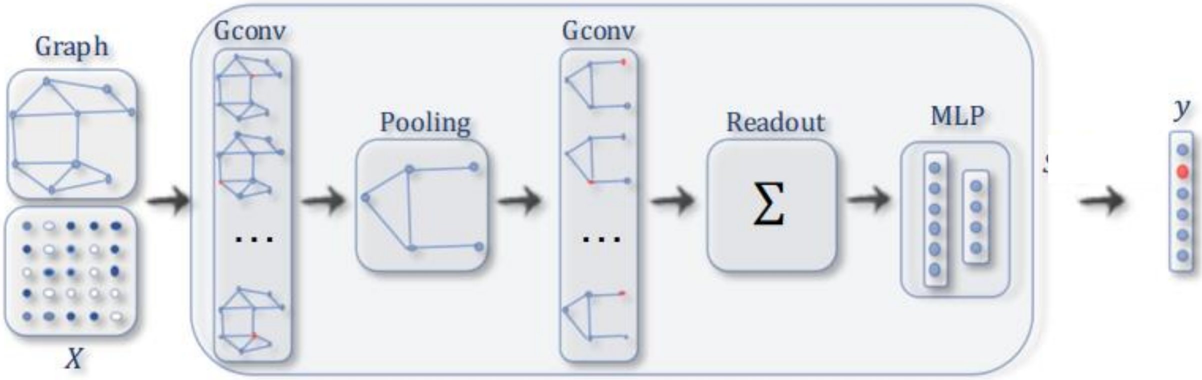


Figure 11: Illustration of GNN

As shown above, the input of the GNN model is one graph associated with its features of individual nodes ***X***.

- **Step 1**: by applying graph convolution layer (Gconv), we can extract different features of the graph centered at different nodes.
- **Step 2**: pooling technique is utilized to down-sample the extracted features. After repeating several times of convolution and pooling on the input graph, we can extract lots of high-level features of the original graph, which are the components of describing the whole graph.
- **Step 3**: readout is used to aggregate all the high-level features into a vector, which is usually equal to sum/mean operation. Up to this step, we have embedded the original graph into a vector.
- **Step 4**: Given the graph embedding vector, we put it into a multi-layer perception module (MLP), which can map the input vector **V** from Euclidean space to our desired space (e.g., failure probability of pipeline). The basic formulation is as $\mathrm{Y} = W^T\mathrm{V} + \mathrm{b}$, where V denotes input embedding vector, Y represents our output, W is learnable parameter with a learnable bias b.

2.1.2. Problem Formulation

Given the preliminary of GNN, we can see that the representation of the graph-based input of is very important. In this section, we will explain how we formulate the interacting threats as graph-based input.

A. Basic definition

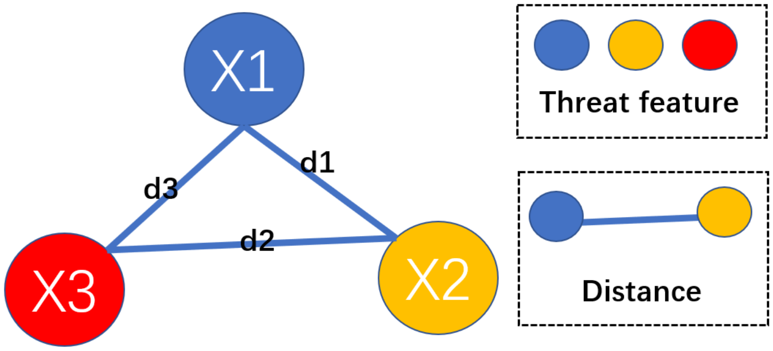


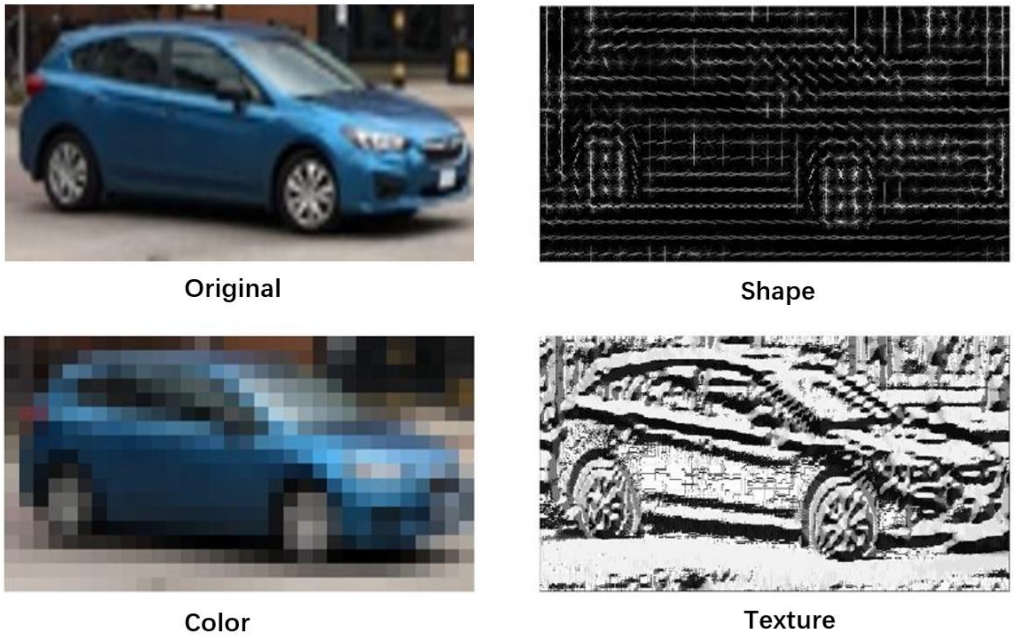
Figure 12: Basic definition of undirect graphs

As shown in the above figure, the interacting threats can be formulated as a undirect graph, which encodes the features of individual threats (e.g., appearance, size, shape, type, etc.) and the distance between a pair of threats. Formally, several interacting threats can be formulated as a graph $\mathrm{G}(\mathbf{X}, \mathbf{D})$, where $\mathbf{X} = \{\boldsymbol{x}_i\}^{\mathrm{n}}$ denoting the node attribute set that contains the m-dimension feature vector $\mathbf{x}_{\mathrm{i}} = [x_1, x_2, \ldots, x_m]$ describing the attribute of individual threats, and $\mathbf{D} = \left\{\mathrm{d}_{\mathrm{j}}\right\}^{\mathrm{k}}$ denoting edge set that represents the distance between pairs of threats. Given the graph-based representation, we take the graphs as input of GNN and do the prediction task. How to generate the node set $\mathbf{X}$ and edge set $\mathbf{D}$ from raw data of interacting threats? We will explain it in the following separately.

B. Generation of node attribute set **X**

Generally, we divide the attributes of individual threats into two categories, including appearance attribute and semantic attribute.

For the appearance attribute of threats, they describe the what the threats look like, including color, shape, texture and the combination of all of them. Since we currently do not have the real threat data, we use real object image as an example to demonstrate what appearance attributes look like.



Original

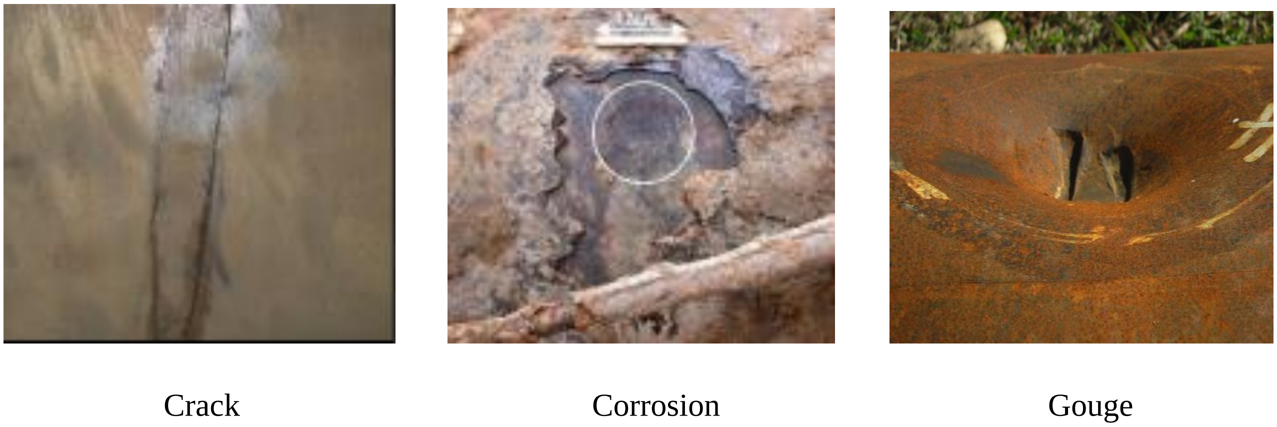
Shape

Color

Texture

Figure 13: Appearance attributes demonstration

As we can see, the appearance attribute can capture the important visual information of threats. For semantic attributes, they represent high-level semantic information of threats, e.g., type, size and so on, as shown below.



Crack

Corrosion

Gouge

Figure 14: Type of threats

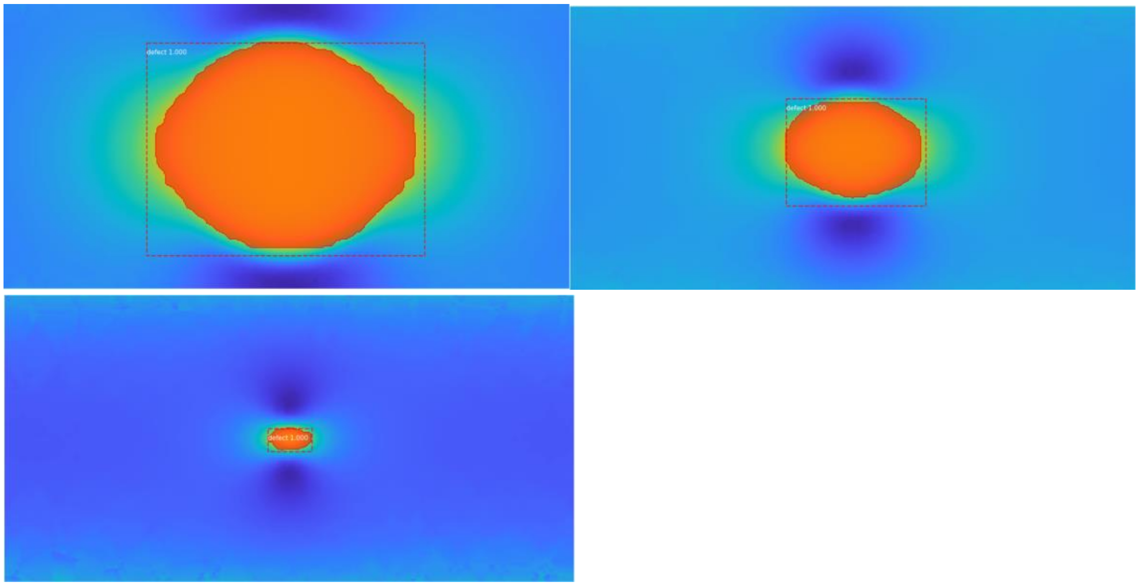


Figure 15: Size of corrosion threats in MFL data

Assume we only have three threats with different types (crack, corrosion and gouge), then we can set $\boldsymbol{x}_1 = 1, \boldsymbol{x}_2 = 2$ and $\boldsymbol{x}_3$=3. If we have three threats with different sizes $s_1, s_2 s_3$, which can be obtained from our previously designed segmentation method, then we can set $\boldsymbol{x}_1 = s_1, \boldsymbol{x}_2 = s_2$ and $\boldsymbol{x}_3$=s_3.

C. Generation of edges set **D**

To generate the edge set **D**, we need to answer three questions, including (1) How to obtain the position of individual threats? (2) How to connect pairs of threats? and (3) How to represent the edges in mathematical form.

For the first question, we can easily obtain the position of the threat in raw MFL data and calculate the distance given the positions, as shown in Figure 6.

For the second question, there are two types of nodes connection, the first type is named fully connected, which means that each node (threat) connects all the other nodes (threat). In this way, all the threats interact with each other. The other type is known as Delaunay triangulation, which generate triangle sets **T** given a set of nodes and ensure that no node is inside the circumcircle of and triangle in the triangle set **T**, as shown below. The Delaunay triangulation can maximum the minimum angle of all the angles of the triangles in T and avoid sliver triangles. In other words, Delaunay triangulation is a sparse way to generate edges among a given set of nodes and only consider the neighborhood nodes as interacting.

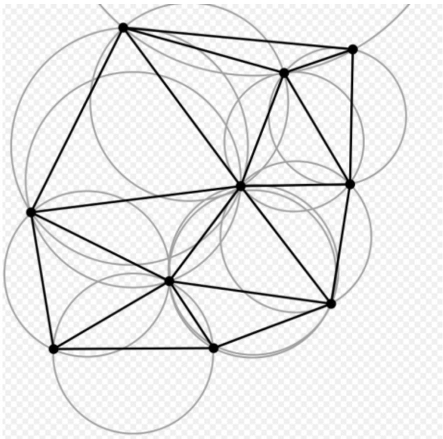
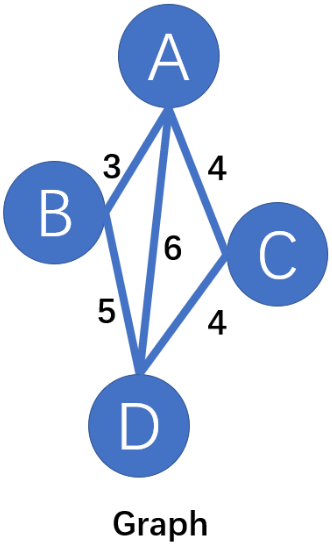


Figure 16: Explanation of Delaunay triangular (from Wiki)

The third question is on the representation of edge set in mathematical form. Generally, we use adjacent matrix to represent the connection and weights of connection among nodes. As shown below, given a novel graph with fixed edge connections and weights, then we can formulate the spatial topology of the graph by the adjacent matrix. If there has no edge connected between two nodes (e.g., node A and node A, or node B and node C), then the corresponding value in adjacent matrix is zero. Otherwise, the value is the weight of the corresponding edge.



	A	B	C	D
A	0	3	4	6
B	3	0	0	5
C	4	0	0	4
D	6	5	5	0

Adjacent matrix

Figure 17: Explanation of adjacent matrix

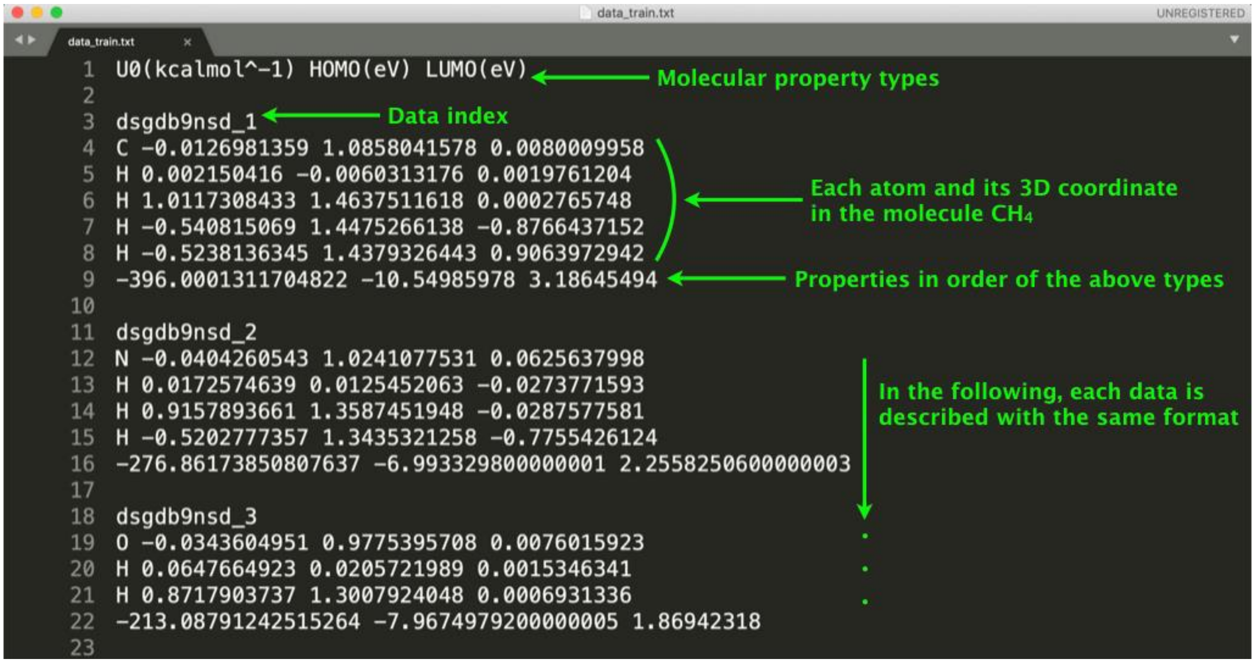
To sum up, we model the interacting threats as a graph with node set and edge set, which represents the features of individual threats and the spatial relationship between a pair of threats separately. Thus, given the graph representation, we can encode the information of individual threats and the interacting threats as the input to deep graph learning model (GNN).

2.2 Dataset used for the evaluation of deep graph learning

We need the dataset of pipeline assessment to train and test our deep graph learning model, which is a **supervised regression** model. Since currently the MSU team is still working on the data generation, and we need to find a replaceable dataset for temporal evaluation. Given our designed supervised regression deep graph learning model, the training dataset should meet three requirements.

- The input of the data can be represented as graphs so that we can mimic the graph-based interacting threats.
- There must exist ground true output given any inputs, which is required by supervised learning model.
- The output value should be continuous, which is required by the regression problem. Since the output we want is failure probability of the pipeline and it is a continuous value, thus, we design the whole learning model as a regression problem, which needs continuous values as output.

Given the above requirements, we decide to use QM9 dataset as a temporal dataset for the training and testing of our designed deep graph learning model, in which the inputs are molecular structures and the predictive outputs are molecular properties. The details are shown below:



```
U0(kcalmol^-1) HOMO(eV) LUMO(eV)

dsgdb9nsd_1
C -0.0126981359 1.0858041578 0.0080009958
H 0.002150416 -0.0060313176 0.0019761204
H 1.0117308433 1.4637511618 0.0002765748
H -0.540815069 1.4475266138 -0.8766437152
H -0.5238136345 1.4379326443 0.9063972942
-396.0001311704822 -10.54985978 3.18645494

dsgdb9nsd_2
N -0.0404260543 1.0241077531 0.0625637998
H 0.0172574639 0.0125452063 -0.0273771593
H 0.9157893661 1.3587451948 -0.0287577581
H -0.5202777357 1.3435321258 -0.7755426124
-276.86173850807637 -6.993329800000001 2.2558250600000003

dsgdb9nsd_3
O -0.0343604951 0.9775395708 0.0076015923
H 0.0647664923 0.0205721989 0.0015346341
H 0.8717903737 1.3007924048 0.0006931336
-213.08791242515264 -7.9674979200000005 1.86942318

```

Molecular property types

Data index

Each atom and its 3D coordinate in the molecule CH_4

Properties in order of the above types

In the following, each data is described with the same format

Figure 18: Details of QM9 dataset

Given the above figure, we can see that QM9 dataset meets all our requirements. We take a bunch of atoms (threats) with 3D coordinate as input, and the output is continuous prediction of molecular properties.

To sum up, we will first use QM9 dataset to evaluate the validation of our designed deep graph learning model. Once the MSU team has finished the data generation, we will transfer our model to the real dataset on pipeline assessment.

3. Summary and Future Work

In this report and the previous report, we completed Task 3.1 (CNN-based data characterization) and Task 4.1 (spatiotemporal graph modeling). In the next quarter, MSU will generate simulation data of interacting threats and time-varied threats, and continue working on CNN-based data characterization and reliability analysis. Mines will implement deep graph learning models based on recurrent neural networks that can model the spatiotemporal relationship of interacting threats to predict the failure probability of pipelines.